Stock Market and Stock News Prediction Prices

Long Short Term Memory models to predict stock prices with stock news data

Adrian Cabreros
California State University of Sacramento
West Sacramento, CA Yolo County
adriancab@gmail.com

Shane Harris
California State University of Sacramento
Sacramento, CA Sacramento County
shane.harris6400@gmail.com

ABSTRACT

Predicting stock market performance is a difficult task. There are many factors that move a stock price up or down which makes the stock market quite volatile and harder to predict with accuracy. However, with growing technology we can attempt to predict stock prices using machine learning as accurately as possible. For this problem we are trying to predict a company's stock price based on its historical performance. We will also be using news articles to leverage our model based on the type of words used in the article. We will also use the sentiment analysis of the articles to help predict stock performance.

We approached this problem by researching how businesses use stock market analysis. Mainly, there are two types of strategies when performing stock market analysis- fundamental analysis and technical analysis

- Fundamental Analysis is the process of analyzing the company's future profitability and value based on the state of the business environment and the financial performance.
- Technical Analysis is the quantitative process of examining and predicting price movements based on historical price charts and market statistics. The idea is to analyze previous market patterns to predict future market trends.

For this problem we will only be focused on the technical analysis. We will be using datasets from trading view and datasets from Stock News API. We then preprocessed the data to fit the model from a date range that dates to around a year from November 2018 - November 2019. We approached the problem by using an LSTM model (Long Short Term Memory). The LSTM is well suited for sequential type datasets since it stores past information in order to predict future sequences. The LSTM has three gates:

- Input gate: adds information to state
- Forget gate: removes information that is no longer needed.

- Output gate: selects information to be show for output.

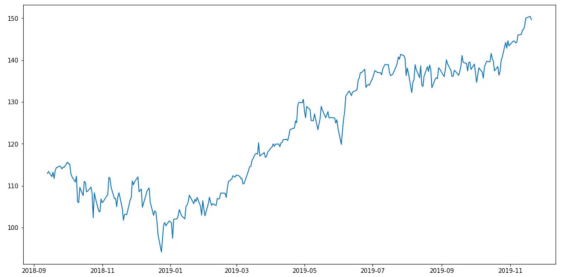
Method

For our method we used an LSTM we gathered data from tradingview.com and stock news api from roughly 45 companies. For the stocknews api we created a script to loop over the api and collect the data.


```
{
    "data": [
        {
            "title": "H&R Block Emerald Advance® line of credit gives consumers access to up to $1,000",
            "news_url": "http://www.globenewswire.com/news-release/2019/11/18/1948742/0/en/
                H-R-Block-Emerald-Advance-line-of-credit-gives-consumers-access-to-up-to-1-000.html",
            "image_url": "https://cdn.snapi.dev/images/v1/r/u/press15.jpg",
            "text": "KANSAS CITY, Mo., Nov.  18, 2019  (GLOBE NEWSWIRE) -- H&R Block knows the unexpected need for cash
                can pop up. Consumers need access to money and Block can provide access to up to $1,000 upon approval
                with the  H&R Block Emerald Advance® line of credit . Starting today consumers can apply at any
                participating H&R Block tax office. Approved applicants are eligible for an unsecured credit limit
                between $350 and $1,000.",
            "sentiment": "Neutral",
            "type": "Article",
            "source_name": "GlobeNewsWire",
            "date": "Mon, 18 Nov 2019 09:00:00 -0500",
            "tickers": [
                "HRB"
            ],
            "topics": [
                "PressRelease"
            ]
```

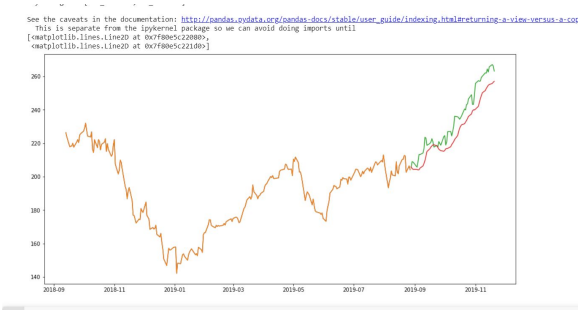

Each request would return 50 objects at most so we had to loop through each page request to gather more data. We would do this 45 times.

For our stock data we used tradingview we requested data from the same 45 companies. Once the data was gathered we preprocessed the data to have the same date formats. We also one hot encoded the categorical data from the news api which was the sentiment of the article, then we vectorized our article texts and title text into a tf-idf format. For the stock data it contained the following attributes: time, open, high, low, close, Volume, Volume MA, Plot, Total, Total BTC, Total ALTs, PE Ratio, PB Ratio, Total Debt, Total Equity, Stock Symbol. We normalized the attributes using min max strategy except the Close price. The order of the data is sequential based on date. For our LSTM we inputed the normalized attributes and set the close price as our outpute are trying to predict.



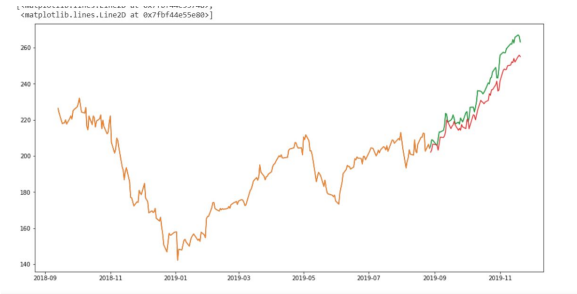
The above Diagram is the stock trend from Microsoft within the past year. We use the first 240 days as our input for the model and try to predict the last 60 days.

After training the model we get the following result. The red line is our predicted result for 60 days and the green is the actual values



This was the result after one epoch and the news data.

We then fit the model using the rest of the companies



Here is our result with Microsoft. As you can see the prediction trend closely resembles the actual trend. As the model is more trained the prediction per day becomes more accurate with the result. We were able to predict the price within a 1.6 price range.

```
close          6.875866
Predictions    5.269640
dtype: float64
```

Our model looks like the following

```
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1],1)))
model.add(LSTM(units=50))
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x_train, y_train, epochs=5, batch_size=1, verbose=2)
```

We transformed our input shape to be three dimensional. Our hidden layers consist of 50 nodes with two layers. Our output layer is 1 since we are trying to predict one value. We used adam as our optimizer. The time for each epoch took about 10-15 seconds.

Results

Our results for predicting future stocks with the news data was quite accurate for our model. When we increased the epochs the accuracy of the model became more accurate on a granular scale. It was able to predict more fluctuations in daily trends much more accurately.

Related Work

We referenced a research paper by two Stanford students that used several models to predict stock prices. They used Logistic Regression, Bayesian Network, Simple Neural Network, and SVM. Their results were not as performant However they did a more granular approach by predict stocks by the minute. They were able to achieve a 92 percent accuracy within a $2 price range. However past $2 Our approach was to predict the end of the day price over a span of a year.

What Was Learned

We learned about a new model that we have not used before in class. The LSTM is an interesting model that works best with sequential data where timing order is important. We also Learned that data preprocessing can be time consuming when the data is unclean. It took us 90 percent of our time on this project cleaning data and making sure it was correctly formatted for model training

References:

[1] Alice Zheng, Jack Jin“Using AI to Make Predictions on Stock Market” , Stanford Research

[2] Aishwayra Singh,Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes)